IN THE SPECIFICATION:

Please replace the third and fourth full paragraphs of specification page 2 with the following replacement paragraphs:

—

FIG. 3 shows a complete bidirectional system using the serializers as in FIG. 1 and ~~deserializer~~deserializers as in FIG. 2. Note that there are eight data lines and a single clock into each serializer and out from each ~~deserializer~~deserializer. The data and clock lines between the serializer and the ~~deserializer~~deserializer are typically differential signals each using two conductors.

The serializer/deserializer~~deserializers~~ of FIG. 3 each contain a PLL that are common in such devices, but PLL's consume significant power, are complex, require long locking times, and occupy considerable chip real estate. It would be advantageous to dispense with PLL's.

—

Please replace the third full paragraph of specification page 5 with the following replacement paragraph:

—

In view of the foregoing background discussion of the prior art, the present invention provides advantages in serializer/deserializer~~deserializer~~ and a method for sending and receiving serial data without using phase of delay locked loops. The serializer/deserializer~~deserializer~~ sends out a data word bit by bit and receives a data word bit

by bit. In preferred applications, the data word bits are sent out with word boundary of framing bits and sometimes with filler bits. The boundary bits help distinguished different data words.

—

Please replace the first, second, third and fourth full paragraphs of specification page 6 with the following replacement paragraphs:

—

The deserializer~~deserializer~~ includes a serial port for receiving a data word bit by bit into a second register for storing the bits. The second register has a data input and a control input. There is a clock signal received with the received serial data bits that defines the received second data word bits. This is used to clock the data into the second register.

In a preferred embodiment, a bi-directional data line and a bi-directional clock line are provided that are buffered from the serializer/deserializer~~deserializer~~ electronics so that the data and clock signal flow directions may be reversed. In a preferred embodiment, a parallel data word is loaded into a shift register by a word or load signal. This might be an addressed write (a load signal) from a computer system parallel bus. A clock shifts the data out over the data line. This shift clock is synchronized with the word clock so that the data is properly loaded before shifting occurs. The shift clock is modified before or after being sent out over the bi-directional clock line coincident and synchronized with the data bits being sent.

In a preferred embodiment, the clock traveling with the data is delayed before or after sending, so that a clock edge or other such signal is present to indicate when a data bit is stable.

In another preferred embodiment, a word load signal, sometimes referred to as a word clock, latches data into buffer registers. Between a serializer and a deserializer~~deserializer~~, the data lines are bidirectional as is the bit clock line. There is an overall master or controller that handles the data and clock direction reversals so that information is not lost.

—

Please replace lines 9 and 10 of specification page 8 with the following replacement lines:

—

FIGS. 1 and 2 are block diagram schematics of a prior art serializer and deserializer~~deserializer~~.

—

Please replace the last full paragraph of specification page 9 with the following replacement paragraph:

—

The computer system interfaces with a serializer/deserializer~~deserializer~~ 90 via a parallel data port or contacts. The serializer 90 serializes the parallel data and sends it to the ~~deserializer~~deserializer/serializer 90,' where that data is de-serialized and read in parallel by the computer system 92.' Data is transferred from computer system 92' to 92 in a similar fashion. In one arrangement there is only a serializer in 90 and only a deserializer~~deserializer~~ in 90.' In this case the data transfer is one way, or simplex. Similarly,

the system may be configured in the opposite direction with only a ~~deserializer~~deserializer in 90 and a serializer in 90.' However, in most applications there will be both a serializer and a ~~deserializer~~deserializer, as shown, in both systems 90 and 90.'

—

Please replace the first and second full paragraphs of specification page 10 with the following replacement paragraphs:

—

The parallel interfaces 96 and 96' have parallel data usually arranged in a bi-directional bus with buffers enabled depending on the direction of the data. The parallel interface also shows two generic control lines, WORD LOAD and RDY. As shown, these lines do not have a direction symbol since they may originate in either the computer systems or the serializer/deserializer~~deserializer~~. In one arrangement, the computer system 92, when RDY is true, will load the next word into the serializer/deserializer~~deserializer~~ with the WORD LOAD signal. However, in another arrangement, the serializer may, after sending out a data word, assert the WORD LOAD signal to strobe in a new parallel word from the computing system 92. The computing system would then use the WORD LOAD to bring up the next word to be serialized.

When the computer system 92 or 92' is receiving a parallel data word from the deserializer~~deserializer~~, the RDY line can be used as an indication that the data word is available over the data lines 96. However, in another arrangement, the RDY signal may be used to strobe the data into the computer system. For example, if the data were going directly into a memory system, the RDY signal may strobe in the data word and also increment an address counter to have the memory pointing to the next location. In the fol-

lowing descriptions, the RDY and the WORD LOAD signal may be described with other acronyms that are easily understood to be functionally equivalent.

—

Please replace the paragraph bridging specification pages 10 and 11 with the following replacement paragraph:

—

In yet another arrangement, the serializer/~~deserializer~~deserializers 90 and 90'output the pulse generation/timing signal 100 and 100.' These pulse signals may be distributed to other serializer/~~deserializer~~deserializers to provide a clock for such systems. In one example, a free running pulse source that is started when powered up may be distributed for sending and receiving data via other serializers/~~deserializer~~deserializers.

—

Please replace the second and third full paragraphs of specification page 11 with the following replacement paragraphs:

—

In addition to the many physical and functional arrangements or forms of the serializer/deserializer~~deserializer~~, pulse signals from a deserializer~~deserializer~~, say in item 90, may be sent to the sending serializer in 90' over the transmission cable 94. Then, the serializer in item 90', acting as a slave, uses the received pulse signals to serialize and send the data back to the deserializer~~deserializer~~.

Of note in FIG. 7, there is no phase or delayed locking circuitry in the serializer/~~deserializer~~deserializers 90 and 90.' Furthermore, there is no reference of other clock or timing signal fed into the serialzer/de-serializers for providing a locking reference. The serializer/~~deserializer~~deserializers have, instead, a pulse generator that is synchronized and/or gated to the sending and receiving of a data word. The inventive operation shown in FIG. 7 employs the pulse generation as discussed herein, but also provides a bit clock traveling with the serialized data between the two systems 90 and 90.' This bit clock typically will have an edge that is delayed (either at the sender or the receiver) from the data edge so that the bit clock defines the stable portion of a data bit. Alternatively, as is known in the art, the receiver may provide the delay so as to load the bits error free.

—

Please replace the paragraph bridging specification pages 11 and 12 with the following replacement paragraph:

—

In operation, there may be many other variations of the operation and uses of the general block diagram of FIG. 7, and the present invention can be used to advantage in these many applications. Inventive implementations include the following independent arrangements: a) one direction serializer; b) one direction ~~deserializer~~deserializer; c) bidirection data lines; d) bidirection bit clock lines; e) unidirectional bit clock lines; f) unidirectional data lines; g) handshaking; h) no handshaking; i) external word load generation; j) internal word load generation; k) free running synchronized pulse generation; l) gated pulse generation; m) starting the serializing by a word load signal; and n) starting the serializing by a change in the data. The description of these implementations as "independent" refers to the ability of those skilled in the art to employ virtually any of the enumer-

ated arrangements without respect to the other arrangements, excepting, of course, where the arrangements are mutually exclusive (e.g. using handshaking or not).

—

Please replace the second full paragraph of specification page 12 with the following replacement paragraph:

—

In FIG. 7, the items 90 and 90' include pulse generation/timing blocks. FIG. 8A and 8B describe two preferred embodiments. FIG. 8A shows a gated ring oscillator. Here, when the EN signal 110 is true, the oscillator will output a series of pulses OUT 112 whose timing depends on the delay time around the loop of inverters 114 and the NAND 116. The hysteresis of the Schmitt trigger inverter 115, although not necessary, helps to ensure proper oscillation at the designed frequency. The enable pulse occurs concurrently when a data word has been input from a computing system or from receiving a deserializer~~deserializer~~ full data word.

—

Please replace the second full paragraph of specification page 13 with the following replacement paragraph:

—

FIG. 8B illustrates a circuit that synchronizes the bit clock pulses to the signal that loads a new word for sending, noting that this signal may originate within the serializer/deserializer~~deserializer~~ or the computing system. The START signal is a signal, for

example from power up, that starts the ring pulse generator running. In this case, compared to that in FIG. 8A, the ring -pulse generator is free running. When a new word is ready for sending, EN1 goes true and the flop 118 is set on the succeeding OUT1 signal. Then, the OUT1 signal is output via NAND 120 to shift out the data bits. The pulses are also counted so that when the full data word and any boundary bits have been sent, the flop 118 is reset. The counter 122 output may also be used to load in the next word ready to be sent.

—

Please replace the paragraph bridging specification pages 13 and 14 with the following replacement paragraph:

—

FIG. 9 shows one preferred embodiment of the invention. Here, a serializer/deserializer~~deserializer~~ 130/132 transfer data between the parallel A[0:7] 134 lines and the differential serial signal lines BDS+ and BDS- (BDS) 136. The parallel data on the data lines 134 are input to the serializer 130 that are then shifted out via the output buffer 138. A clock signal is output via the buffer 142 to accompany and define the serial data. The data lines BDS are bidirectional and data is received over these same lines via the input buffers 140. There is a received clock or pulse signal via the bidirectional clock BCK+/BCK- (BCKS) lines. The received data is input to a ~~deserializer~~deserializer using the received BCKS signals and assembled as a parallel data word in a register. The data is placed on the data lines 134 when to be read by the computing system. The direction of the data and the clock buffers are determined by the direction circuit 144. When EN- is true (low) the DIR drives either BtoA or AtoB signals true thereby determining the data flow. This particular implementation of the present invention replaces the translator circuit shown in FIG. 5, the new circuit uses an edge detection mechanism 141 that ac-

cepts any transition of a data edge as a signal to transfer the data out. If no data edge occurs or is available, the sending system would provide a strobe of load signal to initiate the transfer, for example, from an enable signal. Such signals are known in the art.

—

Please replace the first full paragraph of specification page 14 with the following replacement paragraph:

—

FIG. 10 is an inventive replacement for the prior art circuit of FIG. 6 where data is strobed into a register. The circuit functions in a manner very similar to that of FIG. 9. The primary difference is that there is a strobe signal 148 that loads the parallel data into a register for serializing. That strobe starts a gated clock or provides the synchronization signal to the free running pulse generation circuitry. In either case, the loaded parallel word is shifted out serially via the buffers BDS along with a bit clock BCKS. When a word is being received, the data is received via the input buffers from BDS lines along with a received BCKS signal. The BCKS signal is used to shift the data into the deserializer~~deserializer~~. The data is moved into a holding register 150. At about the same time, a signal CKP is generated and made available to the computing system to indicate that there is a word ready to input. In response thereto, the data is read by the computing system via the buffers 152 leading to the A[0:7] lines.

—

Please replace the paragraph bridging specification pages 15 and 16 with the following replacement paragraph:

—

In the above systems of FIGS. 9 and 10, the data lines (BDS+, BDS-) and the clock out lines (BCKS+, BCKS-) are typically differential pairs as shown, but may be single ended in other preferred embodiments. Line driver buffers 142 and 138 (FIG. 9) for differential pairs are well known in the art. These differential pairs will be referred to as CKS, and DS unless a specific reference is clearly referring to the individual signals. As mentioned above, typically there will be a deserializer~~deserializer~~ packaged with the serializer 76.

—

Please replace the third full paragraph of specification page 16 with the following replacement paragraph:

—

FIG. 13 illustrates a preferred serializer/deserializer~~deserializer~~ pair operating as a master/slave with unidirectional data transfers. One device 340 is arranged in mode #1 with SER/DES signal set high, the device 340 acting as a serializer and master. Item 342 is the slave operating as a deserializer~~deserializer~~ receiver of the data from 344. Device 342 is arranged with SER/DES signal set low. WORD LOAD is a word clock input, or a load signal, synchronized with the clock circuitry 340 that generates a bit clock 344 with an embedded word boundary. The bit clock is received by the slave 342 via the CKS1 port as shown. Register 348 receives parallel data 346 from a processor via DP_M port that is loaded into the register 348. That data is serialized and sent out synchronously

with the bit clock CKSO via the DS line. The CKSO and the DS are arranged so that each edge of the CKSO is used to load data at the slave 342.